

Création d'un système RSS pour vos news avec PHP 4 et 5

par [Baptiste Wicht \(home\)](#)

Date de publication :

Dernière mise à jour :

Ce tutoriel va vous apprendre à créer votre flux RSS pour vos news. Il est destiné à des personnes ayant déjà de bonnes bases en PHP.

- I - Introduction
- II - Structure du fichier XML
- III - Ajouter les news dans le fichier XML
 - III-A - Méthode d'ouverture
 - III-B - Méthode de création du fichier
 - III-C - Méthode d'ajout d'une news
 - III-D - Méthode de sauvegarde
 - III-E - Initialisation du XML
 - III-F - Exemple : Ajout d'une news
 - III-G - Indiquer au navigateur que l'on possède un flux RSS
- IV - Conclusion
 - IV-A - Conclusion
 - IV-B - Autres formats de Syndication
 - IV-C - Remerciements
 - IV-D - Liens

I - Introduction

Tout d'abord, qu'est ce que le RSS ?

C'est un format de syndication (diffusion en quelque sorte), cela va permettre de diffuser des informations (habituellement des news, des articles voire de nouveaux posts dans des forums) et permettre à d'autres personnes de visualiser ces informations sans visiter votre site. Ces informations seront diffusés sous forme de flux XML. On pourra visualiser ces informations avec un lecteur de flux. Certains browsers (IE 7, Firefox, Opera, etc) possèdent un lecteur de flux, de même que certains moteur de recherche (Google, Yahoo) et bientôt système d'exploitation (Windows Vista). Mais il existe aussi des lecteurs indépendants.

Concrètement, un Flux RSS n'est rien d'autre qu'un simple fichier XML. La seule chose qu'il nous incombera donc de faire est de tenir à jour ce flux RSS et de dire au navigateur où est ce flux.

II - Structure du fichier XML

Pour commencer, il va bien sûr falloir nous créer un fichier XML, et il faut bien entendu respecter une certaine structure pour ce fichier. Voilà donc comment se fait un tel fichier :

Structure du fichier XML

```
<?xml version="1.0" encoding="ISO-8859-15" ?>
<rss version="2.0">
  <channel id="xxx">
    <description>Description du Channel</description>
    <link>Un lien vers la page</link>
    <title>Le titre du channel</title>
  </channel>
</rss>
```

Comme vous le voyez, vous avez une balise RSS, la balise racine du fichier, qui contient une balise *channel* (vous pouvez avoir d'autres channels bien entendu). Un channel est en quelque sorte une catégorie, donc rien ne vous empêche si vous le désirez, de faire une catégorie (channel) I dans laquelle vous mettez chaque message important du forum. Chaque balise *channel* contient une balise *description*, une balise *link* et une balise *title* qui, comme ces noms le laissent deviner, renseignent la description, le lien et le titre du channel en question. J'ai juste rajouté un *id* à *channel* pour simplifier ensuite la recherche de cette balise. Vous pouvez aussi mettre d'autres balises à l'intérieur de cette balise *channel*, mais nous allons nous contenter de ceux-là. Voici une liste complète des balises pouvant figurer dans *channel* :

Balises possibles pour le noeud channel

- <title> = Titre du channel
- <link> = Lien vers la page en question
- <description> = Description du channel
- <language> = La langue du channel
- <copyright> = Le copyright du channel
- <managingEditor> = Adresse email de la personne responsable
- <webMaster> = Adresse email du webmaster du site
- <pubDate> = Première publication du channel
- <lastBuildDate> = Dernière publication du channel
- <category> = Catégorie du channel
- <generator> = Programme utilisé pour générer le flux
- <docs> = Documentation du format RSS employé
- <ttl> = Temps avant le prochain rafraîchissement
- <image> = Image affichée avec le channel
- <rating> = La note PICS du channel
- <textInput> = Pour ajouter une zone de saisie de texte
- <skipHours> = L'heures que les lecteurs de flux(agrégateurs) peuvent ignorer
- <skipDays> = Les jours que les lecteurs de flux(agrégateurs) peuvent ignorer

Maintenant, c'est bien beau, on a un *channel*, mais il nous faut mettre quelque chose dedans. Pour cela, il nous faudra aussi respecter une certaine structure :

Structure d'un item

```
<item>
  <title>Titre de l'élément</title>
  <link>Lien vers l'élément en question</link>
  <pubDate>Date de publication de l'élément</pubDate>
  <description>Description de l'élément</description>
  <author>Auteur de la news</author>
</item>
```

Là non plus, ce n'est pas trop compliqué à comprendre. Pour **UN** élément, on a une balise *item*, dans laquelle, on a une balise *title*, *link*, *pubDate* et *description* qui, comme leurs noms l'indiquent, renseignent le titre, le lien vers l'élément, la date de publication et une description de l'élément. Ici encore, on peut mettre d'autres balises dont voici la liste complète :

Balises possibles pour le noeud item

- <title> = Le titre de l'élément
- <link> = Lien vers l'élément en question
- <description> = Description de l'élément
- <author> = Mail de l'auteur de l'élément
- <category> = Catégorie de cet élément
- <comments> = Lien vers une page de commentaires à propos de l'item
- <enclosure> = Les médias attachés à cet objet
- <guid> = Texte qui identifie de manière unique cet item (on met le titre ici d'habitude)
- <pubDate> = Date de publication
- <source> = Channel auquel l'item appartient

Vous avez donc compris qu'à chaque nouvelle news sur votre site, il va falloir ajouter un objet *item* dans le fichier XML. C'est ce qu'on va voir dans les chapitres suivants.

III - Ajouter les news dans le fichier XML

Pour ajouter des news dans le fichier XML, vous avez deux solutions :

- 1 Vous ajoutez chaque news à main dans le fichier XML. Cette solution est valable pour un petit nombre de news à entrer dans le flux.
- 2 Vous créer une méthode qui, à chaque fois que vous ajoutez une news sur le site, ajoute la news dans le fichier XML. Cette méthode est plus complexe à mettre en oeuvre, mais ensuite bien plus pratique, puisqu'il ne reste rien à faire. C'est celle-là que nous allons voir.

Pour manipuler des fichiers XML avec PHP, on peut utiliser soit DOM XML si on a PHP 4 ou alors DOM si on a PHP 5. Etant donné que les deux version peuvent être utilisées, je mettrai à chaque fois les deux versions de chaque code.

Concrètement, voici ce qu'il nous faudra faire :

Méthodes à créer

- Une méthode qui va créer un fichier XML vide ne contenant que le *channel*
- Une méthode qui va ouvrir le fichier XML existant
- Une méthode qui va sauvegarder le fichier XML
- Une méthode qui va ajouter une news : à utiliser à chaque nouvelle news

Si votre site n'a pas encore de news, vous pouvez vous passer de la deuxième méthode.

Admettons maintenant que votre table news se compose ainsi :

Structure de notre table news

```
CREATE TABLE `news` (  
  `ID` int(11) NOT NULL auto_increment,  
  `title` varchar(100) NOT NULL default '',  
  `timestamp` int(12) NOT NULL default '0',  
  `contents` text NOT NULL,  
  `author` varchar(100) NOT NULL default '',  
  PRIMARY KEY `ID` (`ID`)  
);
```

Vous pouvez bien sûr avoir une toute autre structure, vous n'aurez alors qu'à modifier le code en conséquence.

III-A - Méthode d'ouverture

On va maintenant créer la méthode qui ouvrira le fichier XML. Que fait-elle ? Elle ouvre le fichier XML et renvoie une référence au fichier.

Méthode avec PHP4

```
function openXML(){  
  
  // Ouverture du fichier  
  $file = domxml_open_file("CheminDuFichier.xml");  
  
  //On retourne le fichier XML  
  return $file;  
}
```

Méthode avec PHP5

```
function openXML(){  
  
    // Ouverture du fichier  
    $file = new DOMDocument();  
    $file->load("CheminDuFichier.xml");  
  
    //On retourne le fichier  
    return $file;  
}
```

Vous voyez donc que la syntaxe est assez simple pour utiliser du XML et n'est pas si différente entre PHP4 et PHP5. Le véritable changement de DOM est d'être orienté objet.

III-B - Méthode de création du fichier

Nous allons maintenant écrire la méthode qui va initialiser le fichier XML. Il va ajouter le *channel* dans le fichier et les données du *channel*, puis renvoyer une référence au fichier.

Méthode avec PHP4

```
function createXML(){  
  
    // Création du fichier en mémoire  
    $file = domxml_new_doc("1.0");  
  
    // Création du noeud racine  
    $root = $file->create_element("rss"); //On crée l'élément racine  
    $root->set_attribute("version", "2.0"); //On lui ajoute l'attribut version (2.0)  
    $root = $file->append_child($root); //On insère la racine dans le document  
  
    // Création du noeud channel  
    $element_channel = $file->create_element("channel");//On crée un élément channel  
    $element_channel->set_attribute("id", "news"); //On donne un attribut id à notre channel  
    $element_channel = $root->append_child($element_channel);//On ajoute cet élément à la racine  
  
    // Création du noeud description  
    $element_description = $file->create_element("description");//On crée un élément description  
    $element_description = $element_channel->append_child($element_description);//On ajoute cet  
    élément au channel  
  
    // Création du texte pour le noeud description  
    $texte_description = $file->create_text_node("Description du channel"); //On crée un texte  
    $texte_description = $element_description->append_child($texte_description); //On insère ce texte  
    dans le noeud description  
  
    // Création du noeud link et ajout du texte à l'élément  
    $element_link = $file->create_element("link");  
    $element_link = $element_channel->append_child($element_link);  
    $texte_link = $file->create_text_node("Lien vers le site");  
    $texte_link = $element_link->append_child($texte_link);  
  
    // Création du noeud title et ajout du texte à l'élément  
    $element_title = $file->create_element("title");  
    $element_title = $element_channel->append_child($element_title);  
    $texte_title = $file->create_text_node("Titre du Channel");  
    $texte_title = $element_title->append_child($texte_title);  
  
    //On retourne le fichier XML  
    return $file;  
}
```

Méthode avec PHP5

```
function createXML(){

    // Création du fichier en mémoire
    $file = new DOMDocument("1.0");

    // Création du noeud racine
    $root = $file->createElement("rss"); //On crée l élément racine
    $root->setAttribute("version", "2.0"); //On lui ajoute l attribut version (2.0)
    $root = $file->appendChild($root); //On insère la racine dans le document

    // Création du noeud channel
    $element_channel = $file->createElement("channel");//On crée un élément channel
    $element_channel->setAttribute("id", "news"); //On donne un attribut id à notre channel
    $element_channel = $root->appendChild($element_channel);//On ajoute cet élément à la racine

    // Création du noeud description
    $element_description = $file->createElement("description");//On crée un élément description
    $element_description = $element_channel->appendChild($element_description);//On ajoute cet élément
    au channel

    // Création du texte pour le noeud description
    $texte_description = $file->createTextNode("Description du channel"); //On crée un texte
    $texte_description = $element_description->appendChild($texte_description); //On insère ce texte
    dans le noeud description

    // Création du noeud link et ajout du texte à l élément
    $element_link = $file->createElement("link");
    $element_link = $element_channel->appendChild($element_link);
    $texte_link = $file->createTextNode("Lien vers le site");
    $texte_link = $element_link->appendChild($texte_link);

    // Création du noeud title et ajout du texte à l élément
    $element_title = $file->createElement("title");
    $element_title = $element_channel->appendChild($element_title);
    $texte_title = $file->createTextNode("Titre du Channel");
    $texte_title = $element_title->appendChild($texte_title);

    //On retourne le fichier XML
    return $file;
}
```

Je ne vais pas commenter à nouveau tout ce code, puisque les commentaires dans le code devraient suffire à sa compréhension.

Comme vous le voyez la création se passe toujours de manière hiérarchique : on crée un élément, on l'ajoute à la racine, on crée un nouvel élément, on l'ajoute à l'élément parent et ainsi de suite. C'est toujours ainsi que se passe la manipulation de fichiers XML.

III-C - Méthode d'ajout d'une news

On va maintenant créer la méthode qui va nous permettre d'ajouter une news dans notre fichier XML.

Méthode avec PHP4

```
function addOneNews($file, $title, $timestamp, $author){

    //On récupère le channel
    $element_channel = $file->get_element_by_id("news");

    // Création du noeud item
    $element_item = $file->create_element("item");
    $element_item = $element_channel->appendChild($element_item);
```

Méthode avec PHP4

```
// Création du noeud title et ajout du texte à l élément
$element_title = $file->create_element("title");
$element_title = $element_item->append_child($element_title);
$texte_title = $file->create_text_node($title);
$texte_title = $element_title->append_child($texte_title);

// Création du noeud link et ajout du texte à l élément
$element_link = $file->create_element("link");
$element_link = $element_item->append_child($element_link);
$texte_link = $file->create_text_node("Lien vers la news");
$texte_link = $element_link->append_child($texte_link);

// Création du noeud pubDate et ajout du texte à l élément
$element_date = $file->create_element("pubDate");
$element_date = $element_item->append_child($element_date);
$texte_date = $file->create_text_node($date("d/m/Y H:i", $timestamp));
$texte_date = $element_date->append_child($texte_date);

// Création du noeud author et ajout du texte à l élément
$element_author = $file->create_element("author");
$element_author = $element_item->append_child($element_author);
$texte_author = $file->create_text_node($author);
$texte_author = $element_author->append_child($texte_author);
}
```

Méthode avec PHP5

```
function addOneNews($file, $title, $timestamp, $author){

//On récupère le channel
$element_channel = $file->getElementById("news");

// Création du noeud item
$element_item = $file->createElement("item");
$element_item = $element_channel->appendChild($element_item);

// Création du noeud title et ajout du texte à l élément
$element_title = $file->createElement("title");
$element_title = $element_item->appendChild($element_title);
$texte_title = $file->createTextNode($title);
$texte_title = $element_title->appendChild($texte_title);

// Création du noeud link et ajout du texte à l élément
$element_link = $file->createElement("link");
$element_link = $element_item->appendChild($element_link);
$texte_link = $file->createTextNode("Lien vers la news");
$texte_link = $element_link->appendChild($texte_link);

// Création du noeud pubDate et ajout du texte à l élément
$element_date = $file->createElement("pubDate");
$element_date = $element_item->appendChild($element_date);
$texte_date = $file->createTextNode($date("d/m/Y H:i", $timestamp));
$texte_date = $element_date->appendChild($texte_date);

// Création du noeud author et ajout du texte à l élément
$element_author = $file->createElement("author");
$element_author = $element_item->appendChild($element_author);
$texte_author = $file->createTextNode($author);
$texte_author = $element_author->appendChild($texte_author);
}
```

Il n'y a rien de nouveau dans cette partie de code, puisque nous avons déjà traité de l'organisation hiérarchique plus haut.

III-D - Méthode de sauvegarde

On va maintenant passer à la méthode de sauvegarde. On ne crée cette méthode que pour rendre plus pratique notre code, ainsi en passant PHP4 à PHP5 ou inversement, l'ensemble des fonctions DOM et DOMXML se retrouvent dans le fichier php dont on ne fait qu'appeler les méthodes, il suffit donc de changer un fichier RSS.php par l'autre et le tour est joué.

Méthode PHP4

```
function saveXML($file){  
  
    //Sauvegarde du fichier  
    $file->dump_file("CheminDuFichier.xml");  
}
```

Méthode PHP5

```
function saveXML($file){  
  
    //Sauvegarde du fichier  
    $file->save("CheminDuFichier.xml");  
}
```

III-E - Initialisation du XML

Voici maintenant le code qui va vous permettre d'initialiser votre fichier XML :

Initialisation XML

```
require("RSSPHP.php");  
  
//Création du fichier XML  
$file = createXML();  
  
mysql_connect("localhost","pseudo","mdp") or die(mysql_error());  
mysql_select_db("db") or die(mysql_error());  
  
$query_news = mysql_query("SELECT * FROM news") or die(mysql_error());  
  
while($data_news = mysql_fetch_array($query_news)){  
    addOneNews($file,$data_news['title'],$data_news['timestamp'],$data_news['author']);  
}  
  
mysql_close();  
  
saveXML($file);
```

Ce code ne fait que créer le fichier XML et appeler la méthode addOneNews pour chaque news qui est dans la base de données. Vous allez employer normalement une seule fois ce code ; la première fois pour que le XML soit prêt à accueillir de nouvelles News.

III-F - Exemple : Ajout d'une news

Maintenant que l'on a notre beau code qui marche, il faut bien l'utiliser, à moins que vous ne l'ayez fait pour faire joli...

Comme notre exemple fonctionne avec les news, on va faire le code d'insertion d'une news dans la base de données. Disons que cette page est appelée depuis un formulaire, ça nous donnerait un code comme ça :

Utilisation concrète

```
extract($_POST);

require("RSSPHP.php");

mysql_connect("localhost","pseudo","mdp") or die(mysql_error());
mysql_select_db("db") or die(mysql_error());

//Ajout de la news dans la base de données
mysql_query("INSERT INTO news (titre, timestamp, contenu, auteur) VALUES
(' . $title . ',' . $timestamp . ',' . $contents . ',' . $author . ')") or
die(mysql_error());

//Ouverture du fichier
$file = openXML();

//Ajout de la news dans le fichier
addOneNews($file, $title, $timestamp, $author);

saveXML($file);

mysql_close();
```

Vous pouvez constater que pour quelqu'un qui aurait déjà fait son script de news, le changement est minime. Il suffit d'appeler la méthode `addNews` et le tour est joué.

Si c'est la première fois que vous ajoutez le RSS, il ne faut pas oublier AVANT cela d'utiliser le code que je vous ai fourni au chapitre précédent pour que le xml soit déjà créé et le channel initialisé avant que l'on entre des items dedans.

III-G - Indiquer au navigateur que l'on possède un flux RSS

Maintenant que toutes vos news sont directement mises dans le XML et que votre fichier XML est terminé, il faut que vous indiquiez à votre navigateur et aux lecteurs de flux (agrégateurs) où se trouve ce fichier. Pour cela, il vous suffira d'ajouter la ligne suivante dans la balise `head` de votre page :

Balise à ajouter pour indiquer qu'il y a un flux RSS sur votre site

```
<link rel="alternate" type="application/rss+xml" title="Flux RSS" href="Lien vers le fichier XML" />
```

Normalement, si tout s'est bien déroulé et que vous naviguez sur votre site avec Firefox, vous devriez voir ce logo



Logo RSS

s'afficher à côté de l'URL de votre page. Vous pouvez donc dès à présent lire le flux RSS de votre site avec n'importe quel lecteur de flux.

IV - Conclusion

IV-A - Conclusion

Dans cet exemple, nous avons créé un flux RSS pour les news du site, mais on peut aussi imaginer d'y mettre autre chose, tel que les posts d'un forum, les commentaires d'un blog et j'en passe. La manière de faire serait exactement la même, il vous faudra juste l'adapter aux données que vous voudrez mettre dans ce nouveau flux.

IV-B - Autres formats de Syndication

Bien que nous n'ayons traité que RSS, il existe d'autres formats de Syndication, dont le plus connu (sans compter RSS) est Atom. Il est très semblable à RSS, mais vise à être plus flexible, il peut notamment être distribué via le protocole HTTP. Son point fort est surtout son approche plus professionnelle que celle de RSS.

La syntaxe XML d'ATOM est plus compliquée que celle de RSS. En voici un exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title type="text">dive into mark</title>
  <subtitle type="html">
    A <em>lot</em> of effort
    went into making this effortless
  </subtitle>
  <updated>2005-07-31T12:29:29Z</updated>
  <id>tag:example.org,2003:3</id>
  <link rel="alternate" type="text/html"
    hreflang="en" href="http://example.org/">
  <link rel="self" type="application/atom+xml"
    href="http://example.org/feed.atom"/>
  <rights>Copyright (c) 2003, Mark Pilgrim</rights>
  <generator uri="http://www.example.com/" version="1.0">
    Example Toolkit
  </generator>
  <entry>
    <title>Atom draft-07 snapshot</title>
    <link rel="alternate" type="text/html"
      href="http://example.org/2005/04/02/atom/">
    <link rel="enclosure" type="audio/mpeg" length="1337"
      href="http://example.org/audio/ph34r_my_podcast.mp3"/>
    <id>tag:example.org,2003:3.2397</id>
    <updated>2005-07-31T12:29:29Z</updated>
    <published>2003-12-13T08:29:29-04:00</published>
    <author>
      <name>Mark Pilgrim</name>
      <uri>http://example.org/</uri>
      <email>f8dy@example.com</email>
    </author>
    <contributor>
      <name>Sam Ruby</name>
    </contributor>
    <contributor>
      <name>Joe Gregorio</name>
    </contributor>
    <content type="xhtml" xml:lang="en"
      xml:base="http://diveintomark.org/">
      <div xmlns="http://www.w3.org/1999/xhtml">
        <p><i>[Update: The Atom draft is finished.]</i></p>
      </div>
    </content>
```

```
</entry>
</feed>
```

Si vous voulez plus d'infos sur ce format, vous trouverez une introduction à ce format dans les liens en bas cette page, qui devrait vous aider à prendre en main Atom.

IV-C - Remerciements

Je tiens à remercier **Yoqui** pour ses conseils sur la rédaction de cet article.

IV-D - Liens





Vous pouvez télécharger ici le fichier `rss.php` qui contient toutes les méthodes que nous avons codé :

Source de cet article

- [RSSPHP4.php](#)
- [RSSPHP5.php](#)

Voici des liens qui pourraient vous être utile pour la compréhension de ce tutoriel :

Liens intéressants pour aller plus loin

-  [Documentation officielle de DOMXML](#)
-  [Documentation officielle de DOM](#)
-  [Tutoriel sur l'utilisation de DOM](#)
-  [Spécifications RSS 2.0 Officielles](#)
-  [Plus d'infos sur RSS](#)
-  [Introduction à Atom](#)
-  [Spécifications Atom](#)
-  [Une liste non exhaustive d'agrégateurs pour Windows](#)

